IN THE CLAIMS:

Please amend the claims as follows:

1-241. (Cancelled)

242-254. (Withdrawn)

255-262. (Cancelled)

263. (Withdrawn)

264-282. (Cancelled)

283. (Withdrawn)

284. (Cancelled)

285. (Previously Presented) A computer-implemented method for programmatically creating a graphical program, comprising:

storing a first program, wherein the first program is executable to automatically create a new graphical program based on received information;

executing the first program, wherein said executing comprises automatically creating the new graphical program based on the received information, wherein said programmatically creating the new graphical program comprises:

automatically creating a plurality of graphical program nodes in the new graphical program; and

automatically interconnecting the plurality of graphical program nodes in the new graphical program;

wherein the interconnected plurality of graphical program nodes comprise at least a portion of the new graphical program; and

wherein said automatically creating the new graphical program creates the new graphical program without any user input specifying the plurality of graphical program nodes or the interconnection of the plurality of graphical program nodes during said creating.

286. (Previously Presented) The method of claim 285, wherein the new graphical program comprises a diagram portion comprising the plurality of interconnected nodes and a user interface portion; and

wherein said automatically creating the new graphical program comprises creating the diagram portion and the user interface portion.

287. (Previously Presented) The method of claim 285, wherein the new graphical program comprises a data flow diagram.

288. (Previously Presented) The method of claim 285, wherein the new graphical program is a virtual instrument.

289. (Previously Presented) The method of claim 285, wherein said executing the first program occurs in a first computing environment;

wherein said first computing environment is connected to a second computing environment;

wherein said executing the first program comprises sending information from the first computing environment to the second computing environment; and

wherein the new graphical program is created in the second computing environment.

290. (Previously Presented) The method of claim 285,

wherein the first program specifies creation of the new graphical program.

291. (Previously Presented) The method of claim 285, further comprising:

receiving information from a user;

wherein said automatically creating comprises automatically creating the new graphical program at least partially based on the information received from the user.

292. (Previously Presented) The method of claim 285,

wherein the first program comprises a client program that calls an application programming interface (API) to automatically create the new graphical program; and

wherein said executing comprises the first program calling the API to perform said creating and said interconnecting.

293. (Previously Presented) The method of claim 285,

wherein the first program comprises a server program; and

wherein said executing comprises the first program receiving calls from a client program.

294. (Previously Presented) The method of claim 285, wherein the new graphical program comprises a block diagram, and wherein the plurality of graphical program nodes comprise at least one function node placed in the block diagram.

295. (Previously Presented) The method of claim 285, wherein the new graphical program includes a block diagram, and wherein the plurality of graphical program nodes comprise at least one programmatic structure placed in the block diagram.

296. (Previously Presented) The method of claim 285, wherein the plurality of graphical program nodes comprise at least one graphical loop structure.

297. (Previously Presented) The method of claim 285, wherein the plurality of graphical program nodes comprise at least one graphical case structure.

298. (Previously Presented) The method of claim 285, wherein said interconnecting the plurality of graphical program nodes comprises displaying a connection between an input of a first graphical program node and an output of a second graphical program node.

299. (Previously Presented) The method of claim 285, wherein said automatically creating the new graphical program comprises:

automatically creating one or more user interface nodes, wherein the one or more user interface nodes perform one or more of providing input to or displaying output from the new graphical program.

300. (Previously Presented) The method of claim 285, wherein the new graphical program includes a user interface panel, and wherein plurality of graphical program nodes comprise at least one user interface node placed in the user interface panel.

301. (Previously Presented) The method of claim 300, wherein the user interface node comprises at least one of:

a user interface input node placed in the user interface panel for providing user input to the new graphical program; and/or

a user interface output node placed in the user interface panel for viewing output of the new graphical program.

302. (Previously Presented) The method of claim 285, wherein said automatically creating the new graphical program further comprises:

obtaining a reference to a graphical program node, wherein the reference is used to manipulate the graphical program node.

303. (Previously Presented) The method of claim 285, wherein said automatically creating the new graphical program further comprises:

performing at least one of getting or setting a property of a graphical program node.

304. (Previously Presented) The method of claim 285, wherein said automatically creating the new graphical program further comprises:

invoking a method on a graphical program node.

305. (Previously Presented) The method of claim 285, wherein the first program is a first graphical program.

306. (Previously Presented) The method of claim 305, wherein the first graphical program includes at least one object creation node for automatically creating at least one graphical program node in the new graphical program; and

wherein said automatically creating the new graphical program comprises including the at least one graphical program node in the new graphical program.

307. (Previously Presented) The method of claim 306, wherein the first graphical program further includes a property node; and

wherein said automatically creating the new graphical program comprises the property node getting or setting a property of the graphical program node.

308. (Previously Presented) The method of claim 307, wherein the object creation node outputs a reference to the graphical program node;

wherein the property node receives as input the reference to the graphical program node; and

wherein the property node gets or sets a property of the graphical program node specified by the reference to the graphical program node.

309. (Previously Presented) The method of claim 306, wherein the first graphical program further includes an invoke node; and

wherein said automatically creating the new graphical program comprises the invoke node invoking a method on the graphical program node.

310. (Previously Presented) The method of claim 309, wherein the object creation node outputs a reference to the graphical program node;

wherein the invoke node receives as input the reference to the graphical program node; and

wherein the invoke node invokes a method on the graphical program node specified by the reference to the graphical program node.

311. (Previously Presented) The method of claim 309, wherein the invoked method connects the graphical program node to another graphical program node in the new graphical program.

312. (Previously Presented) The method of claim 311, wherein said connecting the graphical program node to said another graphical program node comprises connecting an input of the graphical program node to an output of said another graphical program node.

313. (Previously Presented) The method of claim 309, wherein the invoked method performs one of 1) moving the graphical program node to another location in the new graphical program; and 2) resizing the graphical program node in the new graphical program.

314. (Previously Presented) The method of claim 306, wherein said creating the first graphical program comprises:

displaying the object creation node; and

specifying a graphical program node class for the object creation node;

wherein the at least one graphical program node included in the new graphical program is of the graphical program node class.

315. (Previously Presented) The method of claim 314, wherein said creating the first graphical program further comprises:

specifying a graphical program node sub-class for the object creation node;

wherein the graphical program node included in the new graphical program is of the node sub-class.

316. (Previously Presented) The method of claim 306, wherein said creating the first graphical program comprises:

displaying the object creation node; and

specifying position information to the object creation node;

wherein the at least one graphical program node included in the new graphical program is positioned in the new graphical program at a location based on the position information.

317. (Previously Presented) The method of claim 306, wherein said creating the first graphical program comprises:

displaying the object creation node; and

specifying owner reference information for the object creation node, wherein the owner reference information designates an owner entity;

wherein the at least one graphical program node is included in the new graphical program as a member of the owner entity.

318. (Previously Presented) The method of claim 317, wherein the owner entity is an entity from the group consisting of: 1) the new graphical program; and 2) another graphical program node of the new graphical program.

319. (Previously Presented) The method of claim 305, wherein the first graphical program includes a graphical program creation node for automatically creating the new graphical program.

320. (Previously Presented) The method of claim 319, wherein said creating the first graphical program comprises:

displaying the graphical program creation node; and

specifying a new graphical program type for the graphical program creation node;

wherein said creating the new graphical program comprises creating the new graphical program of the specified new graphical program type.

321. (Previously Presented) The method of claim 320, wherein the graphical program creation node includes a type input; and

wherein said specifying a new graphical program type for the graphical program creation node comprises connecting type information to the type input of the graphical program creation node.

322. (Previously Presented) The method of claim 319, wherein said creating the first graphical program comprises:

displaying the graphical program creation node; and

specifying a reference to a server program for the graphical program creation node;

wherein said creating the new graphical program comprises the server program creating the new graphical program.

323. (Previously Presented) The method of claim 322, wherein the server program is an application instance of a graphical programming development environment..

324. (Previously Presented) The method of claim 322, wherein the graphical program creation node includes a server program reference input; and

wherein said specifying a reference to a server program for the graphical program creation node comprises connecting information specifying a server program to the server program reference input of the graphical program creation node.

325. (Previously Presented) The method of claim 322, wherein said executing the first graphical program is performed in a first computing system;

wherein said server program executes in a second computing system; and

wherein the first computing system is connected to the second computing system.

326. (Previously Presented) The method of claim 305, wherein said creating the first graphical program comprises:

displaying a graphical program creation node, wherein the graphical program creation node is operable to automatically create the new graphical program;

displaying an object creation node, wherein the object creation node is operable to automatically create at least one graphical program node in the new graphical program; and

configuring the object creation node with one or more inputs.

327. (Previously Presented) The method of claim 326, further comprising:

connecting the graphical program creation node to the object creation node;

wherein the graphical program creation node outputs a reference to the new graphical program; and

wherein said connecting the graphical program creation node to the object creation node comprises connecting the reference to the new graphical program to an input of the object creation node.

328. (Previously Presented) The method of claim 326, further comprising configuring the graphical program creation node with one or more inputs, wherein said configuring the graphical program creation node with one or more inputs comprises performing one or more of:

1) specifying a new graphical program type for the graphical program creation node; and 2) specifying a server reference for the graphical program creation node.

329. (Previously Presented) The method of claim 328, wherein a server reference is specified for the graphical program creation node;

wherein said executing the first graphical program comprises executing program instructions on a first computer;

wherein the server reference references a server program running on a second computer;

wherein the second computer is connected to the first computer via a network; and

wherein said creating the new graphical program in response to said executing the first graphical program comprises the server program creating the new graphical program.

330. (Previously Presented) The method of claim 326, wherein said configuring the object creation node with one or more inputs comprises performing one or more of:

1) specifying a node class for the object creation node; 2) specifying a node sub-class for the object creation node; 3) specifying position information to the object creation node; and 4) specifying owner reference information for the object creation node.

331. (Previously Presented) The method of claim 305, wherein the first graphical program includes a plurality of object creation nodes each for automatically creating a graphical program node in the new graphical program, wherein said plurality of object creation nodes includes a first object creation node for creating a first graphical program node in the new graphical program and includes a second object creation node for creating a second graphical program node in the new graphical program;

wherein said executing the first graphical program comprises including the first graphical program node and the second graphical program node in the new graphical program;

wherein the first graphical program further includes a node operable to connect the first graphical program node to the second graphical program node; and

wherein said executing the first graphical program includes connecting the first graphical program node to the second graphical program node.

332. (Previously Presented) The method of claim 305,

wherein the first graphical program includes a graphical program creation node for automatically creating the new graphical program;

wherein the first graphical program includes at least one object creation node for automatically creating at least one graphical program node in the new graphical program; and

wherein said executing the first graphical program includes:

executing the graphical program creation node, wherein said executing the graphical program creation node causes creation of the new graphical program; and executing the object creation node, wherein said executing the object creation node causes inclusion of the graphical program node in the new graphical program.

333. (Previously Presented) A memory medium, comprising:

a first program, wherein the first program is operable to automatically create a new graphical program based on received information;

wherein the first program comprises program instructions that are operable to:

create a plurality of graphical program nodes in the new graphical program; and

interconnect the plurality of graphical program nodes in the new graphical program;

wherein the interconnected plurality of graphical program nodes comprise at least a portion of the new graphical program; and

wherein the first program is operable to automatically create the new graphical program without any user input specifying selection of graphical program nodes and interconnection of graphical program nodes.

334. (Previously Presented) The memory medium of claim 333, wherein the new graphical program comprises a diagram portion comprising the plurality of interconnected nodes and a user interface portion; and

wherein the first program is operable to automatically create the diagram portion and the user interface portion.

335. (Previously Presented) The memory medium of claim 333, wherein the new graphical program comprises a data flow diagram.

336. (Previously Presented) The memory medium of claim 333, wherein the new graphical program is a virtual instrument.

337. (Previously Presented) The memory medium of claim 333, wherein the first program is operable to execute in a first computing environment;

wherein said first computing environment is connected to a second computing environment;

wherein during execution the first program is operable to send information from the first computing environment to the second computing environment; and

wherein the new graphical program is created in the second computing environment.

338. (Previously Presented) The memory medium of claim 333

wherein the first program specifies creation of the new graphical program.

339. (Previously Presented) The memory medium of claim 333,

wherein the first program is operable to receive information from a user and automatically create the new graphical program at least partially based on the information received from the user.

340. (Previously Presented) The memory medium of claim 333,

wherein the first program comprises a client program that calls an application programming interface (API) to automatically create the new graphical program; and

wherein during execution the first program is operable to call the API to perform said creating and said interconnecting.

341. (Previously Presented) The memory medium of claim 333,

wherein the first program comprises a server program; and

wherein during execution the first program is operable to receive calls from a client program.

342. (Previously Presented) The memory medium of claim 333, wherein the new graphical program comprises a block diagram, and wherein the plurality of graphical program nodes comprise at least one function node placed in the block diagram.

343. (Previously Presented) The memory medium of claim 333, wherein the new graphical program includes a block diagram, and wherein the plurality of graphical program nodes comprise at least one programmatic structure placed in the block diagram.

344. (Previously Presented) The memory medium of claim 333, wherein the plurality of graphical program nodes comprise at least one graphical loop structure.

345. (Previously Presented) The memory medium of claim 333, wherein the plurality of graphical program nodes comprise at least one graphical case structure.

346. (Previously Presented) The memory medium of claim 333, wherein, in interconnecting the plurality of graphical program nodes, the program instructions are operable to display a connection between an input of a first graphical program node and an output of a second graphical program node.

347. (Previously Presented) The memory medium of claim 333, wherein the program instructions are further operable to:

create one or more user interface nodes, wherein the one or more user interface nodes perform one or more of providing input to or displaying output from the new graphical program.

348. (Previously Presented) The memory medium of claim 333, wherein the new graphical program includes a user interface panel, and wherein plurality of graphical program nodes comprise at least one user interface node placed in the user interface panel.

349. (Previously Presented) The memory medium of claim 348, wherein the user interface node comprises at least one of:

a user interface input node placed in the user interface panel for providing user input to the new graphical program; and/or a user interface output node placed in the user interface panel for viewing output of the new graphical program.

350. (Previously Presented) The memory medium of claim 333, wherein the program instructions are further operable to:

obtain a reference to a graphical program node, wherein the reference is used to manipulate the graphical program node.

351. (Previously Presented) The memory medium of claim 333, wherein the program instructions are further operable to:

perform at least one of getting or setting a property of a graphical program node.

352. (Previously Presented) The memory medium of claim 333, wherein the program instructions are further operable to:

invoke a method on a graphical program node.

353. (Previously Presented) A memory medium, comprising:

a first graphical program, wherein the first graphical program includes a graphical program creation node for automatically instantiating a new graphical program, wherein the first graphical program also includes a first object creation node for creating a first node in the new graphical program and includes a second object creation node for creating a second node in the new graphical program;

wherein the first graphical program is executable to:

instantiate the new graphical program;

include the first node and the second node in the new graphical program; and

connect the first node to the second node; and

wherein the first graphical program is operable to automatically create the new graphical program without any user input specifying selection of graphical program nodes and interconnection of graphical program nodes.

354. (Previously Presented) The memory medium of claim 353, wherein the first graphical program also includes a node for obtaining a graphical program node reference;

wherein the reference to the existing graphical program is provided to the node for obtaining a graphical program node reference;

wherein the node for obtaining a graphical program node reference is configured to obtain a reference to a particular node of the existing graphical program;

wherein the first graphical program includes a modify node;

wherein the reference to the particular node of the existing graphical program is provided to the modify node; and

wherein the modify node is configured to modify the particular node of the existing graphical program.

355. (Currently Amended) A computer-implemented method for automatically creating a graphical program, comprising:

storing a first program, wherein the first program includes a graphical program creation function for automatically instantiating a new graphical program, wherein the first program also includes an object creation function for automatically including a node in the new graphical program; and

executing the first program, wherein said executing comprises:

automatically instantiating the new graphical program; [[and]]

automatically including the node in the new graphical program; and

automatically connecting the node to at least one other node in the new graphical program;

wherein the first program is operable to automatically create the new graphical program without any user input specifying selection of graphical program nodes and interconnection of graphical program nodes.

356. (Previously Presented) The method of claim 355, wherein the first program is a graphical program;

wherein the graphical program creation function comprises a graphical program creation node; and

wherein the object creation function comprises an object creation node.

357. (Previously Presented) The method of claim 355, wherein the first program is a text-based program.

358. (Previously Presented) The method of claim 357, wherein the graphical program creation function comprises a method call to create the new graphical program; and

wherein the object creation function comprises a method call to create the node.

359. (Previously Presented) The method of claim 357, wherein the text-based program obtains a reference to a software component;

wherein the software component enables the text-based program to perform the method call to create the new graphical program; and

wherein the software component enables the text-based program to perform the method call to create the node.

360. (Previously Presented) The method of claim 359, wherein the software component interfaces with a server program;

wherein the server program receives the method call to create the new graphical program;

wherein the server program creates the new graphical program;

wherein the server program receives the method call to create the node; and

wherein the server program creates the node.

361. (Previously Presented) The method of claim 359, wherein the software component is an ActiveX component.

362. (Previously Presented) A system for automatically creating a graphical program, comprising:

a computer system including a CPU and memory;

wherein the memory stores a first program, wherein the first program specifies creation of a new graphical program, wherein the first program is executable to automatically create the new graphical program;

wherein the CPU is operable to execute the first program to automatically create the new graphical program, wherein, in executing the first program, the CPU is operable to:

create a plurality of graphical program nodes in the new graphical program; and

interconnect the plurality of graphical program nodes in the new graphical program;

wherein the interconnected plurality of graphical program nodes comprise at least a portion of the new graphical program; and

wherein the first program is operable to automatically create the new graphical program without any user input specifying selection of graphical program nodes and interconnection of graphical program nodes.

363. (Previously Presented) The system of claim 362, wherein, in response to said CPU executing the first program, the first program is operable to interface with a server program; and

wherein the server program is operable to automatically create the new graphical program in response to said interfacing.

364. (Previously Presented) The system of claim 363, wherein the server program is an application instance of a graphical programming development environment.

365. (Previously Presented) The system of claim 363, wherein said computer system is a first computer system, the system further comprising:

a second computer system;

wherein the server program executes in the second computer system; and

wherein the first computer system is connected to the second computer system via a network.

366. (Previously Presented) The system of claim 362,

wherein at least one of the plurality of graphical program nodes comprises a structure node.

367. (Previously Presented) The system of claim 362, wherein the first program is a first graphical program.

368. (Previously Presented) The system of claim 367, wherein the first graphical program includes at least one object creation node for automatically creating at least one graphical program node in the new graphical program; and

wherein said creating the new graphical program comprises including the at least one graphical program node in the new graphical program in response to said executing the first graphical program.

369. (Previously Presented) The system of claim 367, wherein the first graphical program includes a graphical program creation node for automatically instantiating the new graphical program.

370. (Previously Presented) A system for automatically creating a graphical program, comprising:

a computer system including a CPU and memory;

a client program executing in the computer system, wherein the client program performs API calls to automatically create a graphical program based on received information; and a server program operable to receive the client program calls to automatically create the graphical program based on the received information, wherein, in automatically creating the graphical program, the server program is executable to:

automatically create a plurality of graphical program nodes in the new graphical program; and

automatically interconnect the plurality of graphical program nodes in the new graphical program, wherein the interconnected plurality of graphical program nodes comprise at least a portion of the new graphical program;

wherein the server program is operable to automatically create the new graphical program without any user input specifying selection of graphical program nodes and interconnection of graphical program nodes.

371. (Previously Presented) The system of claim 370, wherein the server program executes on another computer system, and wherein said another computer system is connected to said computer system via a network.

372. (Previously Presented) The system of claim 370,

wherein the client program performs said calls to automatically create a graphical program by obtaining a reference to a software component and invoking methods of the software component; and

wherein the software component is operable to perform the operations of automatically creating the graphical program.

373. (Previously Presented) The system of claim 370,

wherein the client program performs said calls to automatically create a graphical program by obtaining a reference to a software component and invoking methods of the software component; and

wherein the software component relays the client program calls to the server program.

374. (Previously Presented) The system of claim 370, wherein the server program is a graphical programming environment application.

375. (Previously Presented) The system of claim 370, wherein the client program is a client graphical program;

wherein the client graphical program includes a graphical program creation node for automatically instantiating a new graphical program;

wherein the client graphical program also includes an object creation node for automatically creating a graphical program node in the new graphical program; and

wherein said API calls to automatically create a graphical program comprise calls resulting from executing the graphical program creation node and the object creation node.

376. (Previously Presented) The system of claim 375, wherein the client graphical program further includes a property node for getting or setting a property of the graphical program node.

377. (Previously Presented) The system of claim 375, wherein the client graphical program further includes an invoke node for invoking a method on the graphical program node.

378. (Previously Presented) The system of claim 377, wherein the object creation node is a first object creation node for automatically creating a first graphical program node in the new graphical program;

wherein the graphical program also includes a second object creation node for automatically creating a second graphical program node in the new graphical program; and

wherein the invoked method connects the first graphical program node to the second graphical program node.

379. (Previously Presented) A memory medium comprising a client program for automatically creating a new graphical program, wherein the client program comprises:

a means for automatically instantiating the new graphical program;

a means for automatically creating a node in the new graphical program;

a means for getting or setting properties of the new graphical program or the node; and

a means for automatically invoking methods on the new graphical program or the node;

wherein said means for automatically instantiating the new graphical program are operable to automatically instantiate the new graphical program without any user input specifying instantiation of the new graphical program;

wherein said means for automatically creating the node in the new graphical program are operable to automatically create the node in the new graphical program without any user input specifying creation of the node in the graphical program; and

wherein said means for automatically invoking methods on the new graphical program or the node are operable to automatically invoke methods on the new graphical program or the node without any user input specifying invocation of methods on the new graphical program or the node.

380. (Previously Presented) The memory medium of claim 379, wherein the client program is a graphical program;

wherein said means for automatically instantiating the new graphical program comprises a graphical program creation node;

wherein said means for automatically creating a node in the new graphical program comprises an object creation node;

wherein said means for getting or setting properties of the new graphical program or the node comprises a property node; and

wherein said means for automatically invoking methods on the new graphical program or the node comprises an invoke node.

381. (Previously Presented) A computer-implemented method for automatically creating a graphical program, comprising:

storing a first graphical program, wherein the first graphical program includes a graphical program creation node for automatically instantiating a new graphical program, wherein the first graphical program also includes at least one object creation node for automatically creating at least one graphical program node in the new graphical program;

executing the first graphical program, wherein said executing comprises:

automatically instantiating the new graphical program; and

automatically including the node in the new graphical program;

wherein said first graphical program automatically creating the new graphical program comprises automatically creating the new graphical program without any user input specifying selection of graphical program nodes and interconnection of graphical program nodes; and

wherein said automatically including the node in the new graphical program comprises automatically including the node in the new graphical program without any user input specifying inclusion of the node in the new graphical program.